(1)

SonarLint	SonarQube
1) Sonar Lint works more like a plugin.	1) Sonar Qube has a server associated with it.
2) It supports only in the IDE like IntelliJ, Eclipse and visual studio.	2) SonarQube is a central server that processes which covers full Analyses which needs to be triggered by the various Sonarqube Scanners.
3) It gives instant feedback as you type your code	3) It give a vision of the quality of your complete project code base.
4) It concentrates on what you are writing run time while coding	4) They have only command-line adaptions.

(2) Arrays in Java:

Java is a Data structure, the array, which stores a fixed-sized sequential collection of elements of the same type.

→ An Array is used

data, but it is often more useful to think of an array as a collection of variables of the same type.

→ Array is Index-based, the first element of the array is stored in 0th index, 2nd element is stored on 1st index and so on.

→ In Java, array is an object of a dynamically generated class. Java array inherits the object class.

→ we can also create single D, multi D array

Advantages:

1) code optimization (we can retrive & sort the data efficiently)
2) Random access → we can get any data located at an index position.

Disadvantages:

1) size limit — It stores only the fixed size of elements in the array

Programs on Java:

(1) To print sum of all the items of the array.

```
Public class Sumofarray {
  Public static void main(string[] args) {
  int arr() = new int{1,2,3,4,5};
  int sum=0;
  for (int i=0; i<arr.length ; i++){
    Sum = Sum+arr[i];
  }
    System.out.println("Sum of all elements:"
                        +sum);
  }
}
```

O/p: Sum of all elements: 15.

3) static variable

```
Public class session{
Public static void session (string[] args){
System.out.println("value of PI = "+ Math.PI");
}
}
```

Instance variable:

```
Package mini
Public class multiple
{
 int a=10;
 static  b=20;
 Public void static variable()
 {
    system.out.println(a*b);
 }
 Public static Void Main (string[] args)
{
  multiple.obj = new multiple();
   Obj.Multiple();
}
```

→ If else

```
Public static void main (string[] args) {
  for(int j=1; j<3; j++) {
    if (i==2 && j==2) {
      continue;
    }
    System.out.println (i+" "+j);
  }
}
```

Op:
```
1 1
1 2
1 3
2 1
2 3
3 1
3 2
3 3
```

Switch case

```
Public string to English (int number)
{
  Switch (number) {
  Case 0: return "zero";
  Case 1: return "one";
  case 2: return "two";
```

default : return "many";

}

}

If else

```
public class Example Minnumber {
Public static void main (string [] args) {
    int a = 11;
    Int b = 6;
    int c = minfunction(a, b);
  System.out.println ("Minimum value = " + c");
}
public static int min function (int n1, int n2)
{
 if (n1 > n2)
   return n2;
 else
    return n1;
}
}
```